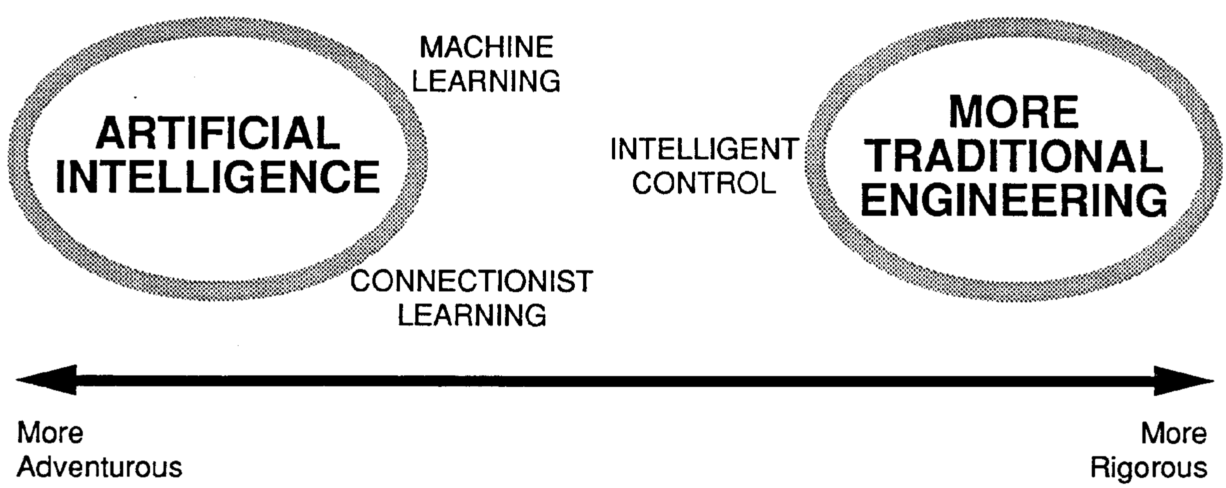
Artificial Intelligence as a Control Problem:
Comments on the Relationship between Machine Learning and Intelligent Control

by Richard S. Sutton
GTE Laboratories Incorporated

Ultimately, the problem of Artificial Intelligence (and thus of Neural Nets) comes down to that of making a sequence of decisions over time so as to achieve certain goals. AI is thus a control problem, at least in a trivial sense, but also in a deeper sense. This view is to be contrasted with AI's traditional view of itself, in which the central paradigm is not that of control, but of *problem solving* in the sense of solving a puzzle, playing a board game, or solving a word problem. Areas where the problem solving paradigm does not naturally apply, such as robotics and vision, have been viewed as outside mainstream AI. I think that the control viewpoint is now much more profitable than the problem solving one, and that control should be the centerpiece of AI and machine learning research.

If both AI and more traditional areas of engineering are viewed as approaches to the general problem of control, then why do they seem so different? In the 1950's and early 1960's these fields were not clearly distinguished. Pattern recognition, for example, was once a central concern of AI and only gradually shifted to become a separate specialized subfield. This happened also with various approaches to learning and adaptive control. I would characterize the split as having to do with the familiar dilemma of choosing between obtaining clear, rigorous results on the one hand, and exploring the most interesting, powerful systems one can think of on the other. AI clearly took the latter "more adventurous" approach, utilizing fully the experimental methodology made possible by digital computers, while the "more rigorous" approach became a natural extension of existing engineering theory, based on the pencil-and-paper mathematics of theorem and proof. See the figure. This is not in any way to judge these fields.



The most striking thing indicated in the figure is not that some work was more rigorous and some more adventurous, but the depth of the gulf between work of these two kinds. Most AI work makes absolutely no contact with traditional engineering algorithms, and vice versa. Perhaps this was necessary for each field to establish its own identity, but now it is counterproductive. The hottest spot in both fields is the one between them. The current enormous popularity of neural networks is due at least in part to its seeming to span these two—the applications potential of rigorous engineering approaches and the enhanced capability of AI. Intelligent control is also in this position.

My conclusion then is that there is indeed a very fruitful area that lies more or less between Intelligent Control and Machine learning (including connectionist or neural net learning), and which therefore presents an excellent opportunity for interdisciplinary research.

VISIT…

LANZAROTE Caliente.com

Dynamic Programming and A* *search.* These two techniques have long been known to be closely related, if not identical. Nevertheless, the complete relationship remains obscure. More importantly, many results have been obtained independently for each technique. How many of these results carry over to the other field? Amazingly, such inter-relations remain almost completely unexplored, at least in the open literature.

Back-propagation. Back-propagation is a connectionist (neural net) learning technique for learning real-valued nonlinear mappings from examples, that is for nonlinear regression (see Rumelhart, Hinton & Williams, 1986). Such a function has many possible uses in control—for learning nonlinear control laws, plant dynamics and inverse dynamics. The important thing is not backpropagation as a particular algorithm—it's clearly limited and will probably be replaced in the next few years—but the idea of a general structure for learning nonlinear mappings. This will remain of relevance to intelligent control.

Temporal-Difference Learning. This is a kind of learning specialized for predicting the long-term behavior of time series. It was first used in a famous early AI program, Samuel's checker player (Samuel, 1959), and since has been used in Genetic Algorithms (Holland, 1986) and in adaptive control in the role of a learned "critic" (Barto, Sutton & Anderson, 1983; Werbos, 1987). The basic idea is to use the change or temporal difference in prediction in place of the error in standard learning processes. Consider a sequence of predictions ending in a final outcome, perhaps a sequence of predictions about the outcome of a chess game, one made after each move, followed by the actual outcome. A normal learning process would adjust each prediction to look more like the final outcome, whereas a temporal-difference learning process would adjust each prediction to look more like the prediction that follows it (the actual outcome is taken as a final prediction for this purpose). If the classic LMS algorithm is extended in this manner to yield a temporal-difference algorithm, then, surprisingly, the new algorithm both converges to better predictions and is significantly simpler to implement (Sutton, 1988).

The Perfect Model Disease. Ron Rivest has coined this term to describe an "illness" that AI (and, to a lesser extent, control theory) has had for many years and is only now beginning to recover from. The illness is the assumption and reliance upon having a perfect model of the world. In toy domains such as the blocks world, puzzle solving, and game playing this may be adequate, but in general of course it is not. Without a perfect model, everything becomes much harder—or at least much different—and so we has been reluctant to abandon the perfect model assumption. The alternative is to accept that our models of the world will always be incomplete, inaccurate, inconsistent, and changing. We will need to maintain multiple models, at multiple levels of abstraction and granularity, and at multiple time scales. It is no longer adequate to view imperfections and inconsistencies in our models as transients and to perform steady-state analysis; we must learn to work with models in which these imperfections will *always* be present. This means certainty equivalence approaches are not enough and dual control approaches are needed.

Control without Reference Signals. The dogma in control is to assume that some outside agency specifies a desired trajectory for the plant outputs in such a way that controls or control adjustments can be determined. For many problems, however, this is simply not appropriate. Consider a chess game. The goal is clearly defined, but in no sense does one ever have a desired trajectory for the game or the moves to be made. Suppose I want a robot to learn to walk bipedally. Producing target trajectories for the joint angles and velocities is a large part of the problem, a part which needs to addressed by learning, not just by analysis and a priori specification. In my opinion, most real control problems are of this sort—in most cases it is natural to provide a specification of the desired result that falls far short of the desired trajectories usually assumed in conventional and adaptive control. This problem will become more and more common as we begin to consider imperfect and weak models, and particularly for systems with long-delayed effects of controls on goals. *Reinforcement learning* represents one approach to this problem (Mendel & McLaren, 1970; Sutton, 1984).